

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111
TELEPHONE (801) 533-9800
FAX (801) 328-1707

TELECOPIER COVER SHEET

February 26, 2010

Total Number of Pages
(including cover letter): 11

Please deliver the transmitted facsimile pages to:

EXAMINER JAMES D. RUTTEN
GROUP ART UNIT: 2192

Telecopier Phone: (571) 273-3703
From: Michael B. Dodd/Connie Boulton
Serial No.: 09/817,880
Comments: Please see attached
Docket No.: 13768.1336
Customer No.: 47973

We are transmitting from a Sharp FO-750 or Sharp FO-6100 facsimile machine. If you do not receive all the pages or they are unreadable, please contact me as soon as possible at (801) 533-9800.

THE INFORMATION CONTAINED IN THIS FACSIMILE MESSAGE IS ATTORNEY PRIVILEGED AND CONFIDENTIAL INFORMATION INTENDED ONLY FOR THE USE OF THE INDIVIDUAL OR ENTITY NAMED ABOVE. IF THE READER OF THIS MESSAGE IS NOT THE INTENDED RECIPIENT, OR THE EMPLOYEE OR AGENT RESPONSIBLE TO DELIVER IT TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISSEMINATION, DISTRIBUTION OR COPYING OF THIS COMMUNICATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE IMMEDIATELY NOTIFY US BY TELEPHONE, AND RETURN THE ORIGINAL MESSAGE TO US AT THE ABOVE ADDRESS VIA THE U.S. POSTAL SERVICE. THANK YOU.

CRB0000000484V001

OK to Enter
/jdr/

Proposed Claims for Examiner's Amendment

Serial No.: 09/817,880

Attorney Docket No.: 13768.1336

Filed: March 26, 2001

**SYSTEM AND METHOD PROVIDING ON-DEMAND GENERATION OF
SPECIALIZED EXECUTABLES**

Claims 1 - 19. (Cancelled).

20. (Currently Amended) A computer implemented method for optimizing image execution that balances competing requirements of the higher performance of native compilation and the development time benefits provided by intermediate language systems, comprising the following computer executable acts:

an execution engine in ~~[[the]]~~a virtual execution environment receiving an intermediate language image;

the execution engine determining ~~[[the]]~~an appropriate execution mode for executing the functionality represented by the intermediate language image, the appropriate execution mode selected from among (a) Just-In-Time processing of the intermediate language image within the virtual execution environment and (b) execution of a compiled native code executable on an underlying platform providing the virtual environment, including:

the execution engine searching a native image repository ~~for to determine if the underlying platform is valid for executing any previously compiled native code executable[[s]] corresponding to the intermediate language image that match environment characteristics of the underlying platform, the native image repository storing previously compiled native code executables for a plurality of different platforms, [[the]]each~~ previously compiled native code executable[[s]] stored along with tags indicating environment characteristics for one of the plurality of different platforms for which the native code executables ~~are~~ is valid, each native code executable stored in the image repository having been previously specialized from some intermediate language image based on specified environment characteristics for native execution on [[a]]the one of the plurality of different platforms for which it is valid ~~having specified environment~~

characteristics;

when the search by the execution engine ~~determining~~ determines that a matching native code executable compiled for native execution on the underlying platform is not available in the native image repository one of the plurality of different platforms for which a previously compiled native code executable is valid[[;]]:

the execution engine processing the intermediate language image in accordance with Just-In-Time processing ~~in response to determining that a native code executable for execution on the underlying platform is not available in the native image repository;~~ and

when the search by the execution engine determines that the underlying platform is one of the plurality of different platforms for which a previously compiled native code executable is valid:

the execution engine accessing the previously compiled native code executable that is valid for the underlying platform from the native image repository; and

the execution engine executing the accessed previously compiled native code executable on the underlying platform in lieu of processing the intermediate language image in accordance with Just-In-Time execution.

21. (Currently Amended) The method of claim 24, further comprising:

a native image generator accessing the intermediate language image and the environment characteristics of the underlying platform from the native image log;

the native image generator ~~generating~~ compiling a specialized native code executable for native execution on the underlying platform, the specialized native code executable generated from the intermediate language image based on the environment characteristics of the underlying platform; and

the native image generator storing the compiled specialized native code executable in the native image repository.

22. (Currently Amended) The method of claim 21, further comprising subsequent to the native image generator storing the specialized native code executable in the native image repository: [[.]]

the execution engine again receiving the intermediate language image;

the execution engine again searching the native image repository to determine if the underlying platform is valid for executing any ~~for previously stored compiled~~ native code executables corresponding to the intermediate language image ~~that match environment characteristics of the underlying platform;~~

the execution engine determining that the underlying platform is valid for executing the specialized native code executable ~~is available for native execution on the underlying platform;~~

the execution engine accessing the specialized native code executable from the native image repository; and

executing the specialized native code executable on the underlying platform in lieu of processing the intermediate language image in accordance with Just-In-Time processing.

23. (Currently Amended) The method of claim 21, ~~further~~ wherein generating compiling the specialized native code executable comprising processing the intermediate language image utilizing standard compilation techniques.

24. (Currently Amended) The method of claim 20, further comprising, when the search by the execution engine determines that the underlying platform is not one of the plurality of different platforms for which a previously compiled native code executable is valid, the execution engine logging the intermediate language image along with environment characteristics of the underlying platform into a native image log, the environment characteristics of the underlying platform obtained during processing of the intermediate language image in accordance with the Just-In-Time processing, the native image log providing input to a native image generator that generates native code executables from intermediate language images and corresponding environment characteristics, the native code executables for native execution on corresponding platforms .

Claims 25 - 33. (Cancelled).

34. (Currently Amended) A computer program product for use at a computer system, the computer program product for implementing a method for optimizing image execution that balances competing requirements of the higher performance of native compilation and the development time benefits provided by intermediate language systems, the computer program product comprising one or more computer storage media having stored thereon computer-executable instructions that, when executed at a processor, cause an execution engine in a[[the]] virtual execution environment to perform the method, including the following:

receive an intermediate language image;

determine [[the]]an appropriate execution mode for executing the functionality represented by the intermediate language image, the appropriate execution mode selected from among (a) Just-In-Time processing of the intermediate language image within the virtual execution environment and (b) execution of a compiled native code executable on an underlying platform providing the virtual environment, including:

searching a native image repository for to determine if the underlying platform is valid for executing any previously compiled native code executable[[s]] corresponding to the intermediate language image that match environment characteristics of the underlying platform, the native image repository storing previously compiled native code executables for a plurality of different platforms, [[the]]each previously compiled native code executable[[s]] stored along with tags indicating environment characteristics for one of the plurality of different platforms for which the native code executables are is valid, each native code executable stored in the image repository having been previously specialized from some intermediate language image based on specified environment characteristics for native execution on [[a]]the one of the plurality of different platforms for which it is valid~~having specified environment characteristics~~;

when the search determines ~~determining that a matching native code executable compiled for native execution on the underlying platform is not~~ one of the plurality of different platforms for which a previously compiled native code executed is valid~~available in the native image repository~~;

processing the intermediate language image in accordance with

~~Just-In-Time processing in response to determining that a native code executable for execution on the underlying platform is not available in the native image repository; and~~

when the search determines that the underlying platform is one of the plurality of different platforms for which a previously compiled native code executable is valid:

accessing the previously compiled native code executable that is valid for the underlying platform from the native image repository; and

executing the accessed previously compiled native code executable on the underlying platform in lieu of processing the intermediate language image in accordance with Just-In-Time execution.

35. (Currently Amended) The computer program product of claim 34, further comprising, when the search by the search engine determines that the underlying platform is not one of the plurality of different platforms for which a previously compiled native code executable is valid, computer executable instructions that, when executed, cause the ~~computer system~~ execution engine to log the intermediate language image along with environment characteristics of the underlying platform into a native image log, the environment characteristics of the underlying platform obtained during processing of the intermediate language image in accordance with the Just-In-Time processing, the native image log providing input to a native image generator that generates native code executables from intermediate language images and corresponding environment characteristics, the native code executables for native execution on corresponding platforms.

36. (Currently Amended) The computer program product of claim 35, further comprising computer executable instructions that, when executed, cause ~~the~~ a native image generator at the computer system to:

access the intermediate language image and the environment characteristics of the underlying platform from the native image log;

~~generate-compile~~ a specialized native code executable for native execution on the underlying platform, the specialized native code executable generated from the

intermediate language image based on the environment characteristics of the underlying platform; and

store the compiled specialized native code executable in the native image repository.

37. (Currently Amended) The computer program product of claim 36, further comprising computer executable instructions that, when executed, cause the execution engine ~~computer system~~ to subsequent to the native image generator storing the specialized native code executable in the native image repository:

receive the intermediate language image again;

search[[ing]] the native image repository again to determine if the underlying platform is valid for executing any previously stored ~~compiled~~ native code executables corresponding to the intermediate language image again;

determine that the specialized native code executable is available for native execution on the underlying platform;

access the specialized native code executable from the native image repository;

and

execute the specialized native code executable on the underlying platform in lieu of processing the intermediate language image in accordance with Just-In-Time processing.

38. (Currently Amended) The computer program product of claim 36, wherein computer executable instructions that, when executed, cause the ~~computer system~~ the native image generator to generate ~~compile~~ the specialized native code executable comprise computer executable instructions that, when executed, cause the computer system to process the intermediate language image utilizing standard compilation techniques.

39. (Currently Amended) A computer system, the computer system including a platform that provides a virtual environment for executing intermediate language images, the computer system comprising:

- a processor;
- system memory;

- a native image log, the native image log entries for one or more intermediate language images, each entry including image information for an intermediate language image and corresponding environment characteristics of the platform where the intermediate language image was executed, the environment characteristics of the platform obtained during processing of the intermediate language image in accordance with Just-In-Time processing;

- a native image repository, the native image repository storing previously compiled native code executables for a plurality of different platforms, the native image repository containing each previously compiled native code executable[[s]] stored along with tags indicating environment characteristics for one of the plurality of different platforms for which the native code executables are is valid, each native code executable stored in the image repository having been previously specialized from some intermediate language image based on specified environment characteristics for native execution on [[a]]the one of the plurality of different platforms for which it is valid having specified environment characteristics;

- one or more computer storage media having stored thereon computer-executable instructions representing an execution engine, a native image service, and a native image generator, wherein the execution engine is configured to:

- receive an intermediate language image;

- determine the appropriate execution mode for executing the functionality represented by the intermediate language image, the appropriate execution mode selected from among (a) Just-In-Time processing of the intermediate language image within the virtual execution environment and (b) execution of a compiled native code executable on the platform, including:

- searching the native image repository to determine if the underlying platform is valid for executing any for a previously compiled

native code executable corresponding to the intermediate language image
~~that match environment characteristics of the platform;~~

when the search determines that the underlying platform is not one of the plurality of different platforms for which a ~~no~~ previously compiled native code executable is ~~found~~valid:

process the intermediate language image in accordance with Just-In-Time processing;

obtain environment characteristics of the platform during processing of the intermediate language image in accordance with Just-In-Time processing; and

log the image information for the intermediate language image along with environment characteristics of the platform into a native image log;

when the search determines that the underlying platform is one of the plurality of different platforms for which a previously compiled native code executable is ~~found~~valid:

access the previously compiled native code executable from the native image repository; and

execute the previously compiled native code executable on the platform in lieu of processing the intermediate language image in accordance with Just-In-Time processing;

wherein the native image service is configured to:

read the image information and environment characteristics for intermediate language images from the native image log; and

forward the image information and environment characteristics to the native image generator; and

wherein the native image generator is configured to:

receive the image information and environment characteristics from the native image service; and

compile a specialized native code executable for native execution on the underlying platform based on the image information and environment

characteristics; and

store the specialized native code executable in the native image repository.

Claim 40. (Cancelled).